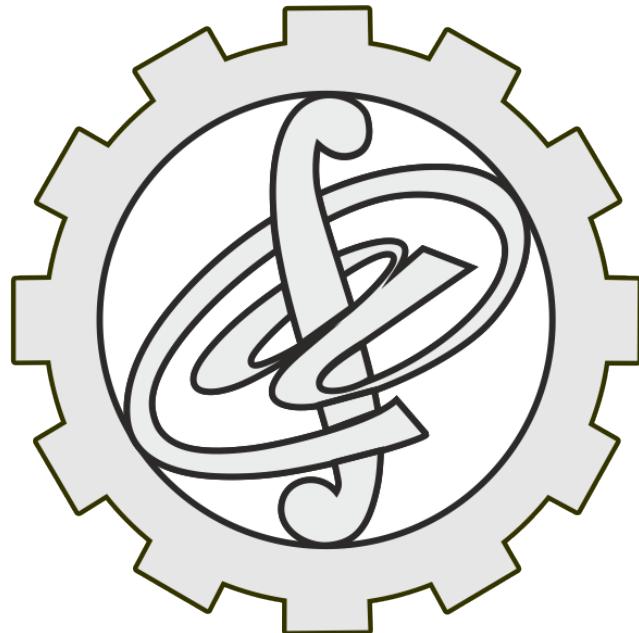


Grafika komputerowa i przetwarzanie obrazów

dokumentacja projektu programu graficznego

Patryk Gamrat, Informatyka I stopnia, semestr 5, grupa 2
Politechnika Śląska, Wydział Matematyki Stosowanej

28 stycznia 2025



Spis treści

1 Wstęp	3
1.1 Opis projektu	3
1.2 Cel programu	3
2 Opis projektu	4
2.1 Funkcjonalności programu	4
2.2 Funkcjonalności dodatkowe	6
2.3 Opis techniczny zaimplementowanych funkcjonalności	7
2.4 Opis zastosowanych funkcji, algorytmów, bibliotek	8
3 Prezentacja projektu	9
3.1 Zrzuty ekranu	9
3.2 Testy programu	11
4 Podsumowanie i wnioski	16
4.1 Podsumowanie	16
4.2 Wnioski	16

1 Wstęp

1.1 Opis projektu

Projekt zaliczeniowy z przedmiotu “Grafika komputerowa i przetwarzanie obrazów” to program graficzny umożliwiający tworzenie prostych grafik, edycję zdjęć, oraz filtrację obrazów z użyciem filtrów. Program posiada kilkadziesiąt narzędzi, oraz możliwość eksportu utworzonych plików. Program został stworzony dla systemu Windows z wykorzystaniem framework'a **WPF** i języka **C#**

1.2 Cel programu

Główym celem stworzonego programu było zapoznanie się z możliwościami utworzenia własnego programu graficznego z wykorzystaniem języka **C#**, a także możliwościami filtracji obrazów z wykorzystaniem biblioteki **EmguCV**. Implementacja programu była możliwa dzięki wykorzystaniu zbioru gotowych klas, narzędzi oraz bibliotek. Szczególnie przydatna okazała się tutaj kontrolka **Canvas**, która jest obszarem roboczym w naszym programie.

2 Opis projektu

2.1 Funkcjonalności programu

Utworzona aplikacja posiada wiele funkcjonalności. Poniżej opisane zostały najważniejsze z nich.

Przybornik z narzędziami

Przybornik z narzędziami umożliwia wybór narzędzia, za pomocą którego chcemy edytować płótno.

Narzędzia dostępne w programie:

- **Pędzel** - pozwala na rysowanie na obszarze roboczym w sposób dowolny poprzez przeciąganie myszki
- **Punkt** - umożliwia nanoszenie pojedynczych punktów
- **Linia** - umożliwia nanoszenie prostych linii poprzez kliknięcie w punkcie początkowym a następnie w punkcie koncowym
- **Edytuj linie** - umożliwia edycję utworzonych wcześniej linii prostych, zostają one podświetlone kolorem niebieskim i można je przeciągać myszką w celu zmiany położenia punktu startowego i końcowego
- **Linia łamana** - umożliwia tworzenie lini łamanych z dowolnej ilości punktów, w tym celu użytkownik musi przytrzymać przycisk CTRL a następnie za pomocą myszki nanosić kolejne punkty
- **Elipsa** - umożliwia tworzenie kształtów elipsy, dodatkowym atutem jest możliwość dowolnego dostosowania kształtu przytrzymując lewy przycisk myszy i przeciągając kształt w dowolnym kierunku (działanie podobne jak w innych popularnych edytorach graficznych)
- **Koło** - działa identycznie jak elipsa, z tą różnicą, że przy rozszerzaniu kształtu wymuszone jest zachowanie proporcji koła
- **Prostokąt** - umożliwia tworzenie dowolnych prostokątów, kształt można dostosowywać tak jak w przypadku elips i kół
- **Kwadrat** - działa identycznie jak prostokąt, z tą różnicą, że przy rozszerzaniu kształtu wymuszone jest zachowanie proporcji kwadratu
- **Trójkąt** - pozwala tworzyć trójkąt, w przypadku tego i kolejnych narzędzi rozszerzanie kształtu nie jest możliwe
- **Sześciokąt** - pozwala tworzyć sześciokąty
- **Strzałka** - pozwala tworzyć strzałki

- **Choinka** - pozwala tworzyć choinki świąteczne, w przypadku tego narzędzia utworzony obiekt składa się z kilku kształtów, oraz posiada predefiniowane kolory
- **Gumka** - umożliwia usuwanie niechcianych elementów z obszaru roboczego

Okno wyboru koloru

W przyborniku z narzędziami znajduje się podgląd aktualnie wybranego koloru. Po kliknięciu w niego, otworzone zostanie okno zmiany koloru. Możliwy jest wybór dowolnego koloru, podając jego wartość w modelu **RGB** lub **HSV**. Po zmianie dowolnej wartości w jednym z modeli przestrzeni barw, wartości zostają automatycznie przekonwertowane na odpowiadające im wartości w drugim modelu. Podgląd koloru jest aktualizowany na bieżaco. Po wyborze odpowiedniego koloru, zostaje on zaaplikowany dla wszystkich narzędzi dostępnych w programie.

Funkcje dostępne z poziomu menu programu

Poniżej opisane są funkcje dostępne z poziomu menu znajdującego się w górnej części okna programu.

Narzędzia dostępne z poziomu menu:

- **Plik -> Zapisz plik** - umożliwia zapisanie zawartości obszaru roboczego do pliku graficznego, po wyborze tej opcji zostanie otworzone okno, gdzie możemy wybrać miejsce zapisu pliku, dostępne formaty plików: .png, .jpg, .gif, .bmp, .tiff i inne
- **Plik -> Wczytaj plik** - umożliwia wczytanie dowolnego pliku graficznego do obszaru roboczego programu w celu jego edycji, dostępne formaty plików są takie same jak w przypadku zapisu
- **Edytuj -> Wyczyść półtno** - całkowicie czyści zawartość obszaru roboczego
- **Edytuj -> Rozmiar półtna...** - umożliwia zmianę rozmiaru obszaru roboczego na dowolną wartość podaną w pikselach, przy zmianie rozmiaru obszaru roboczego jego zawartość jest automatycznie usuwana
- **Filtr** - dostęp do filtrów aplikowanych z użyciem biblioteki EmguCV
- **Pomoc -> O Programie** - wyświetla okno z informacjami dotyczącymi programu

Edycja obszaru roboczego za pomocą filtrów

Program umożliwia edycje aktualnej zawartości obszaru roboczego z użyciem filtrów, funkcjonalność ta jest możliwa dzięki bibliotece EmguCV.

Filtryle dostępne w programie:

- **Sobel** - filtr przydatny do wykrywania krawędzi
- **Canny** - filtr, który również przydaje się do wykrywania krawędzi ale daje odrobine inny efekt w porównaniu z filtrem Sobel
- **Czarno-biały** - filtr, który przekształca obraz do skali szarości
- **Odwroć kolory** - filtr, który odwraca kolory w obszarze roboczym
- **Filtr macierzowy** - otwiera okno, które pozwala zdefiniować własny filtr macierzowy 3x3 i zaaplikować go na obszar roboczy

Filtr macierzowy

W przypadku wybrania filtra macierzowego, otworzone zostanie okno w którym możemy zdefiniować własny filtr macierzowy o rozmiarze 3x3. Możliwe jest wpisanie dokładnej wartości w każdej komórce macierzy. Dodatkowo możemy określić czy chcemy znormalizować filtr przed użyciem, oraz czy zastosować go w skali szarości.

Zastosowanie filtra w skali szarości może być przydatne w przypadku niektórych filtrów np. do detekcji krawędzi.

2.2 Funkcjonalności dodatkowe

Oprócz opisanych powyżej funkcjonalności, program został rozszerzonych o kilka dodatkowych funkcji, które nie były wymagane w założeniach projektu. Poniżej zostały one opisane.

Zaimplementowane funkcje dodatkowe:

- Możliwość zmiany rozmiaru narzędzi w przyborniku - po wyborze odpowiedniego rozmiaru, wszystkie narzędzia stosują ten rozmiar więc możemy tworzyć np. grubsze linie lub kształty z grubszym obramowaniem
- Zapisywanie oraz odczytywanie plików graficznych w różnych formatach - domyślnie wymagane było zapisywanie jedynie w formatach .jpg oraz .png, utworzony program obsługuje też dodatkowe formaty jak np. .gif, .gif albo .tiff
- Obsługa przeźroczystych grafik - o ile sam program nie pozwala ustawić przeźroczystego tła, to możliwe jest jednak wczytanie grafiki z przeźroczystym tłem oraz jej edycja i zapisanie
- Funkcja czyścienia płótna - możliwe jest wyczyszczenie całego obszaru roboczego jednym kliknięciem bez potrzeby restartowania programu

- Funkcja zmiany rozmiaru płótna - program umożliwia ustawienie dowolnego rozmiaru płótna, wpisując szerokość i wysokość w pikselach w osobno otwieranym oknie
- Podpowiedzi dla użytkownika - po wyborze niektórych narzędzi np. do edycji linii, na dole okna wyświetlane są podpowiedzi dotyczące korzystania z wybranego narzędzia

2.3 Opis techniczny zaimplementowanych funkcjonalności

System wyboru narzędzi

Każdy przycisk w przyborniku z narzędziami wywołuje metodę, która sprawdza jakie narzędzie zostało wybrane. Aktualnie wybrane narzędzie jest przechowywane jako wartość Enum. W zależności od wybranego narzędzia wydarzenia odpowiedzalne za wciśnięcie lub poruszenie myszką wywołują inne czynności. Dodatkowo w zależności od wybranego narzędzia, zmianie ulega kurSOR.

Funkcja edycji linii

W przypadku wyboru funkcji edycji lini, wywoływana jest metoda, która przechodzi przez wszystkie elementy w zbiorze roboczym i podświetla linie możliwe do edycji. Po najechaniu na linie możliwą do edycji, kurSOR użytkownika zmienia się. Przeciąganie myszy z wcisniętym przyciskiem sprawia, że pozycja lini aktualizuje się w czasie rzeczywistym.

Funkcja dostosowywania kształtów

Funkcja dostosowywania kształtów takich jak np. elipsa lub prostokąt działa w bardzo prosty sposób. Po wciśnięciu klawiszów myszki, nowy element zostaje utworzony z początkowym domyślnym rozmiarem. Dopóki klawisz myszy jest przytrzymywany, wydarzenie odpowiedzialne za poruszenie myszką rozszerza lub kurczy nowy element do pozycji kurSORa.

Zapisywanie i odczytywanie plików

Zapisywanie i odczytywanie plików jest możliwe dzięki wykorzystaniu klasy SaveFileDialog oraz OpenFileDialog. Po wyborze formatu pliku do zapisu, tworzony jest odpowiedni enkoder, który jest wykorzystywany do zapisania pliku z odpowiednim formatem na podstawie bitmapy obszaru roboczego. W celu zapisu pliku wymagane jest również zastosowanie wielu klas pomocniczych takich jak np. RenderTargetBitmap, VisualBrush, DrawingVisual i DrawingContext.

Funkcja filtrów

Funkcja aplikowania filtrów na obszar roboczy wykorzystuje bibliotekę EmguCV. Po wyborze odpowiedniego filtra, tworzony jest tymczasowy plik graficzny z zawartością obszaru roboczego. Na plik graficzny aplikowany jest odpowiedni filtr. Po wprowadzeniu zmian, nowa grafika zastępuje poprzednią w obszarze roboczym. Plik tymczasowy jest usuwany po skończeniu wszystkich operacji.

2.4 Opis zastosowanych funkcji, algorytmów, bibliotek

Algorytm konwersji pomiędzy modelami RGB i HSV

do konwersji pomiędzy modelami barw RGB i HSV wykorzystywany jest algorytm konwersji. Wartości w oknie do wyboru koloru są aktualizowane na bieżąco, dzięki czemu możliwy jest podgląd koloru w czasie rzeczywistym.

Poniżej znajduje się sposób konwersji pomiędzy modelami barw wykorzystany w projekcie.

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$C_{max} = \max(R', G', B')$$

$$C_{min} = \min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

Hue calculation:

$$H = \begin{cases} 0^\circ & , \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \text{mod}6 \right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , C_{max} = B' \end{cases}$$

Rysunek 1: Formuła konwersji z modelu barw RGB na HSV

Normalizacja filtrów

Przed zastosowaniem filtru macierzowego, zdefiniowanego przez użytkownika, możliwa jest jego normalizacja. Normalizacja polega na tym, że jeżeli suma wszystkich elementów w macierzy nie równa się 0, to każdy element w macierzy dzieli się przez sumę wszystkich elementów. Zastosowanie normalizacji filtra jest konieczne, aby po zastosowaniu filtra nie doprowadzić do zniekształcenia obrazu.

Użyte biblioteki

Wszystkie biblioteki zewnętrzne wykorzystane w projekcie są dostępne z poziomu menadżera pakietów NuGet.

Biblioteki użyte w projekcie:

- **EmguCV** - odpowiednik biblioteki OpenCV dla środowiska .NET, umożliwia wykonywanie szeregu przydatnych operacji na grafikach, w projekcie został wykorzystany do aplikowania filtrów na obszarze roboczym

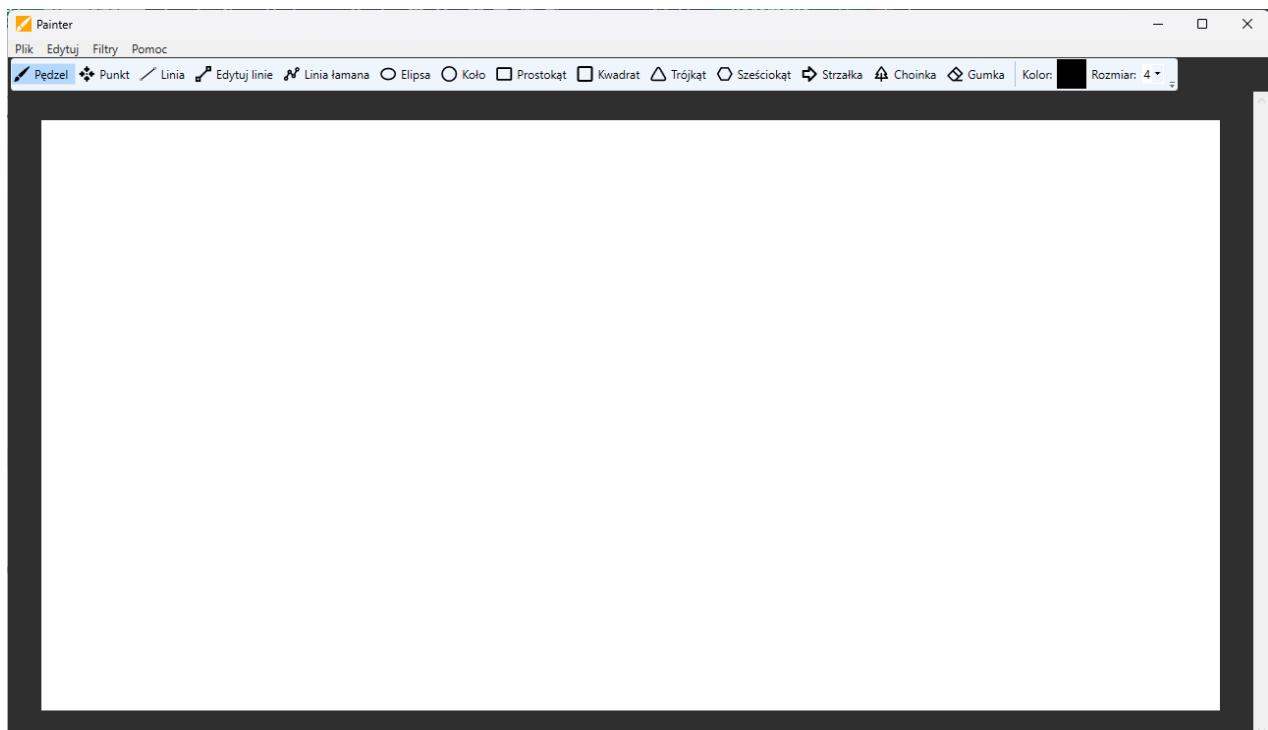
Link do biblioteki: <https://github.com/emgucv/emgucv>

- **MahApps.Metro.IconPacks** - biblioteka zawierająca ogromny zbiór darmowych ikon do wykorzystania, pozwala w łatwy sposób dodawać ikony do kontrolek WPF, w projekcie ikony z tej biblioteki wykorzystywane są w przyborniku z narzędziami oraz jako ikona programu

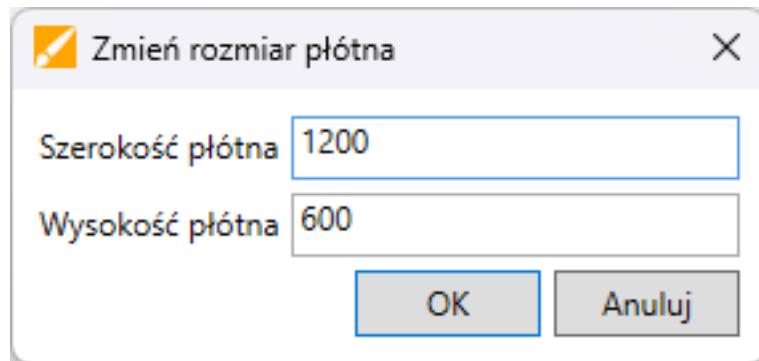
Link do biblioteki: <https://github.com/MahApps/MahApps.Metro.IconPacks>

3 Prezentacja projektu

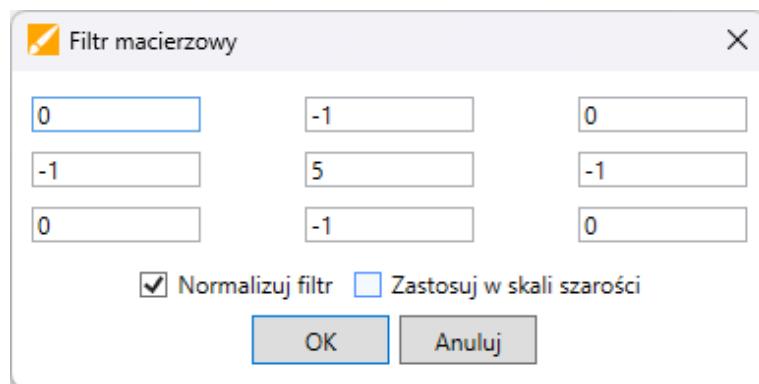
3.1 Zrzuty ekranu



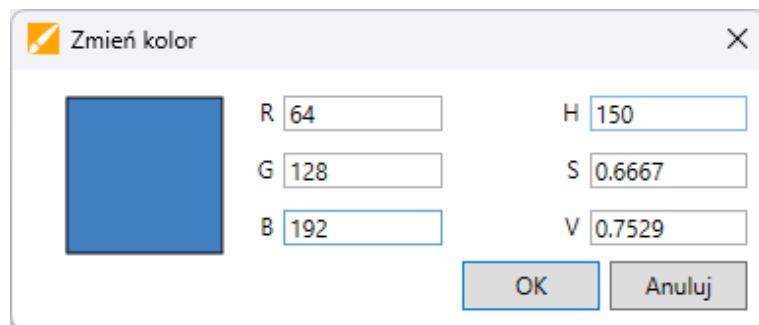
Rysunek 2: Główne okno programu



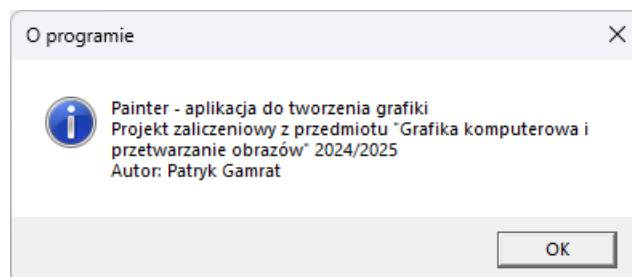
Rysunek 3: Okno zmiany rozmiaru płótna



Rysunek 4: Okno dla filtrów macierzowych

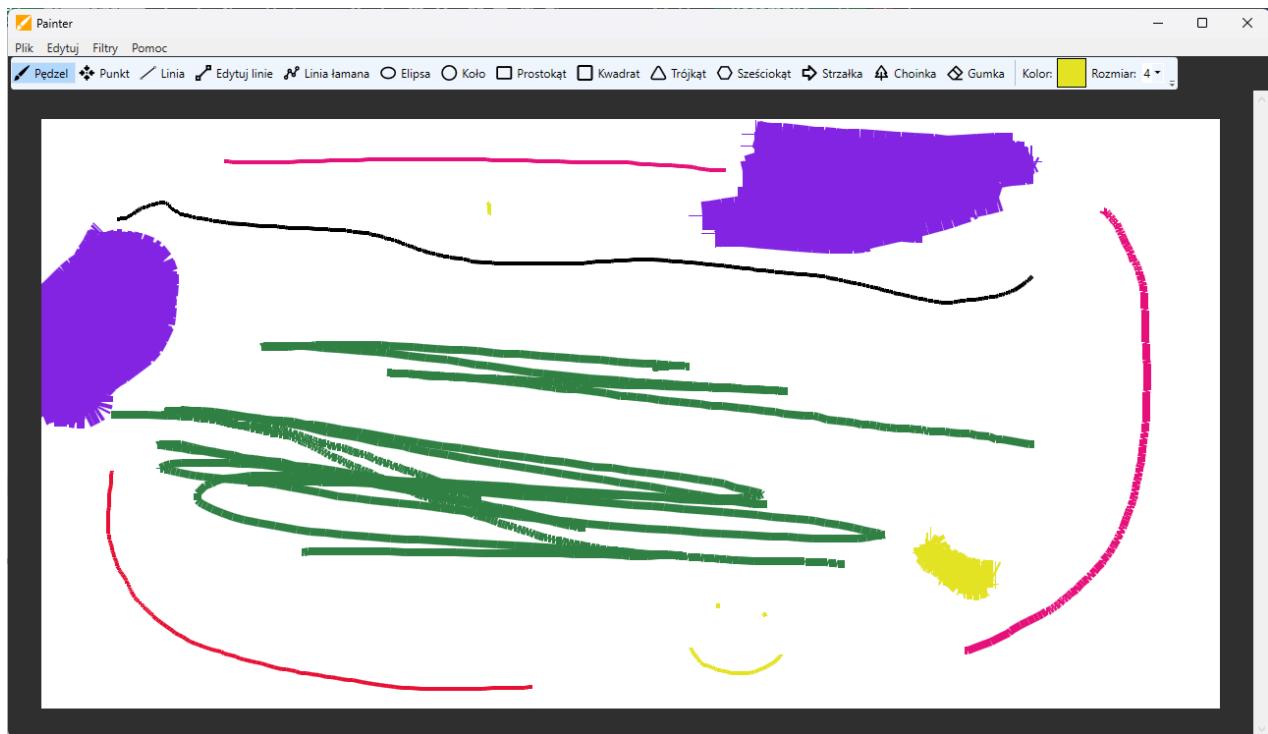


Rysunek 5: Okno wyboru koloru

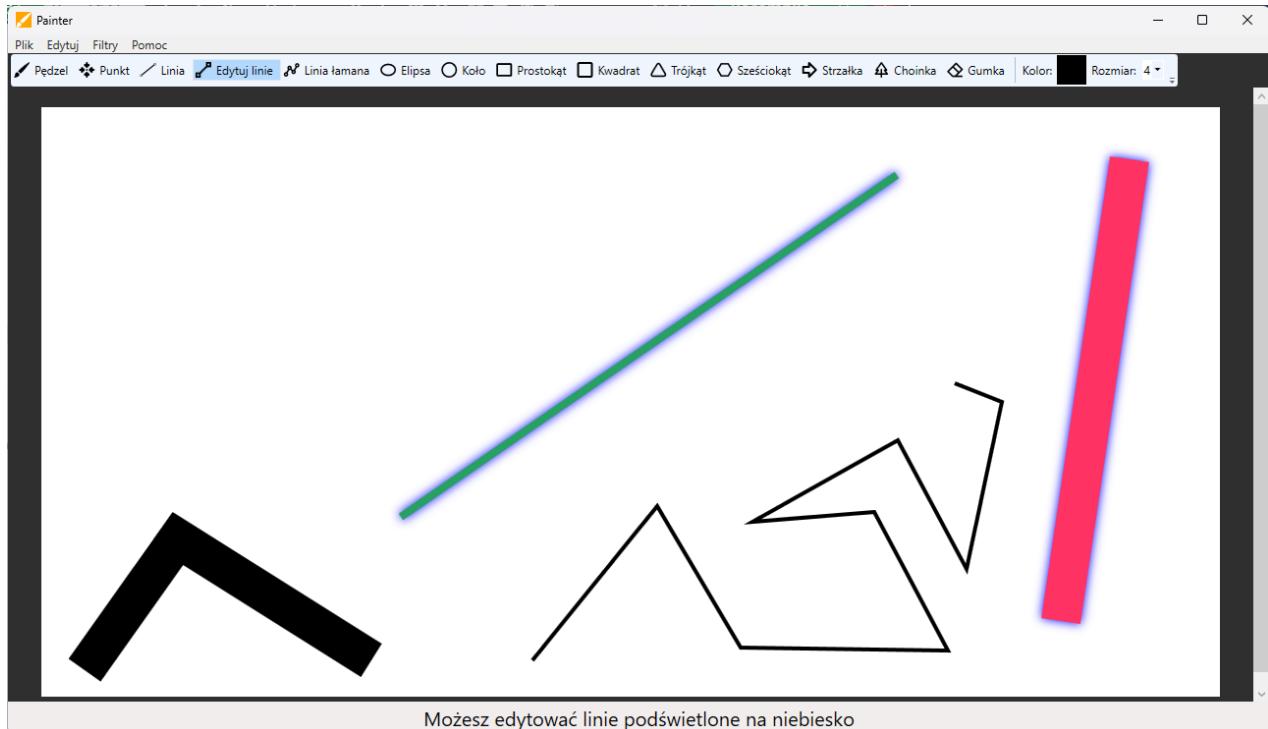


Rysunek 6: Okno z informacjami o programie

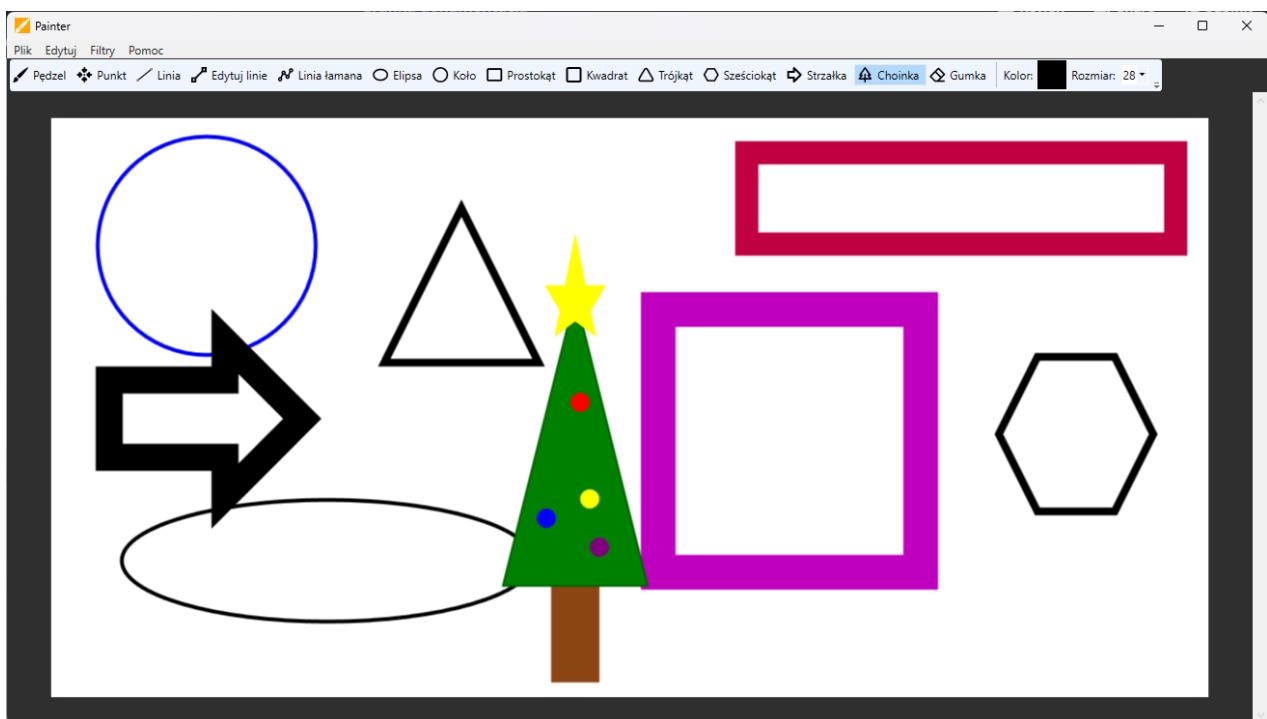
3.2 Testy programu



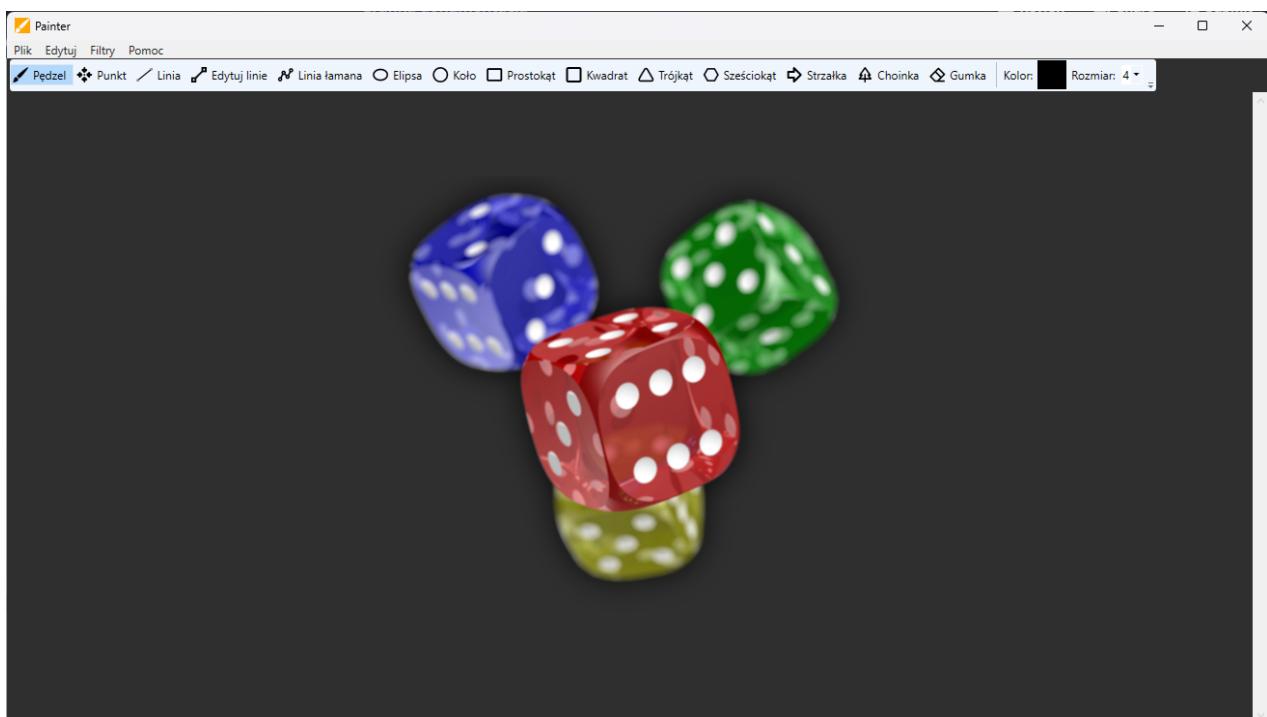
Rysunek 7: Test narzędzia pędzla z wykorzystaniem różnych kolorów i rozmiarów



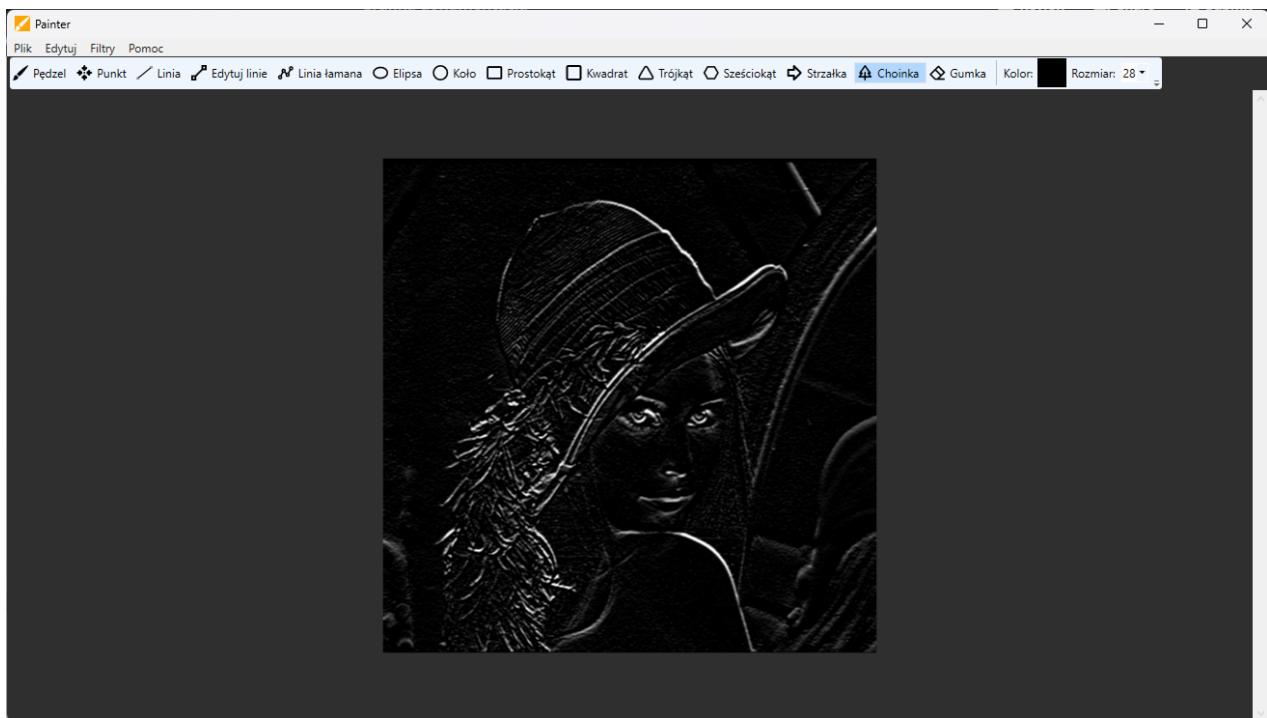
Rysunek 8: Test narzędzia lini prostych, lini łamanych oraz edycji linii



Rysunek 9: Test narzędzi do tworzenia kształtów

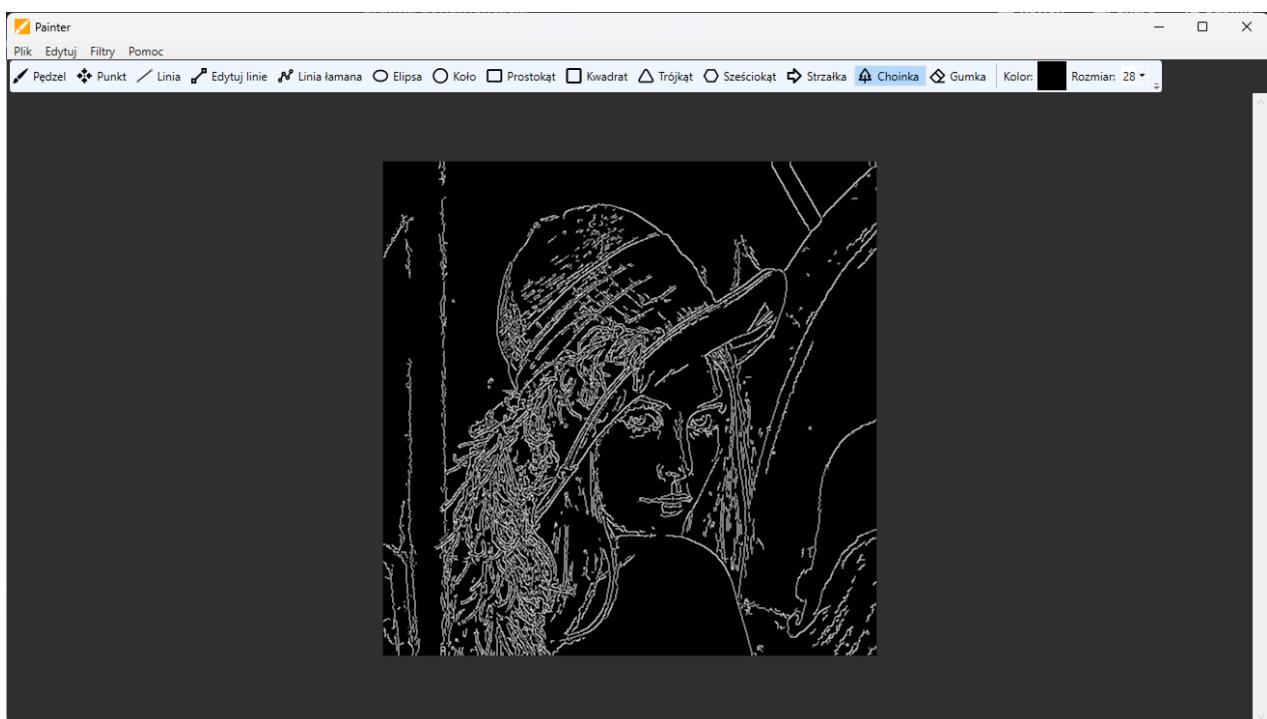


Rysunek 10: Test obsługi zdjęć przeźroczystych



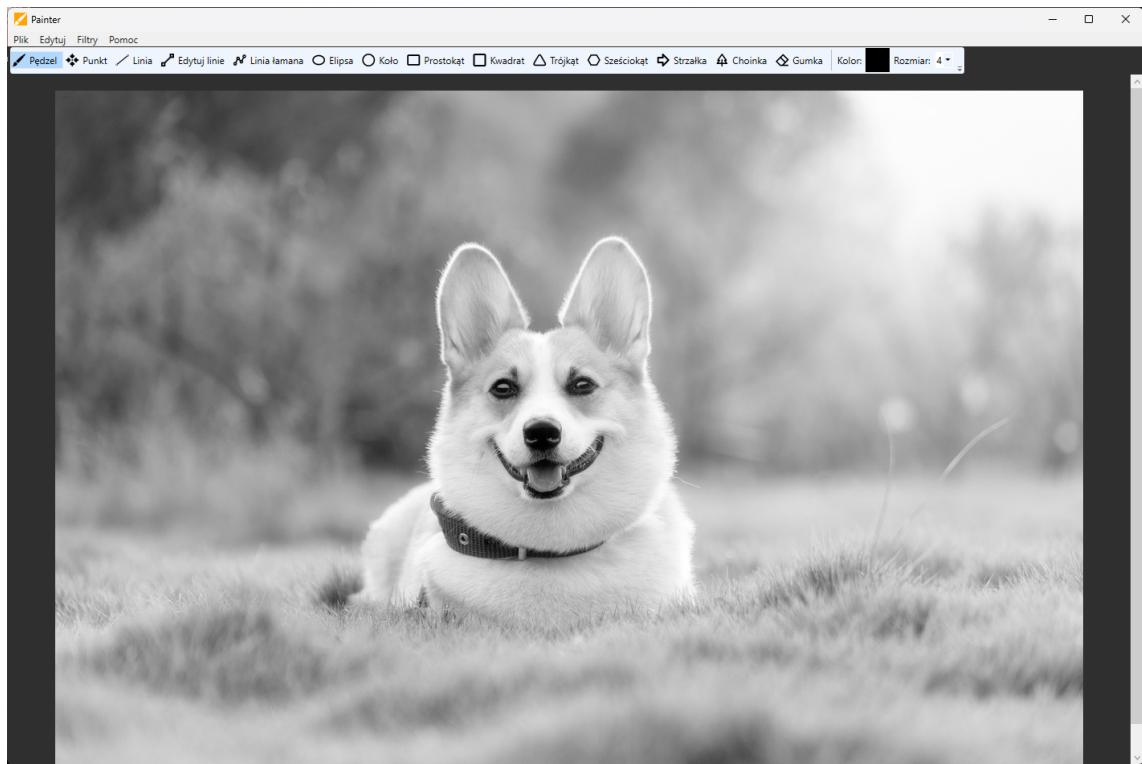
Rysunek 11: Test filtrów - sobel

Grafika: [en.wikipedia.org/wiki/Lenna#/media/File:Lenna_\(test_image\).png](https://en.wikipedia.org/wiki/Lenna#/media/File:Lenna_(test_image).png)



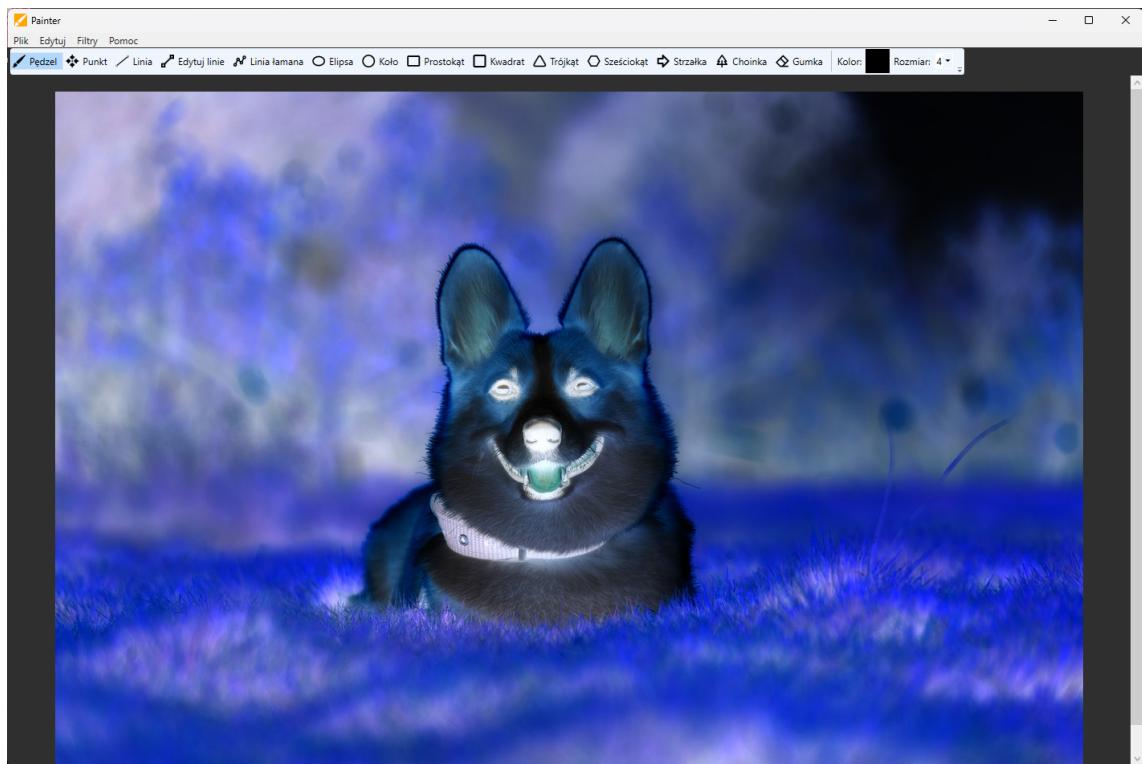
Rysunek 12: Test filtrów - canny

Grafika: [en.wikipedia.org/wiki/Lenna#/media/File:Lenna_\(test_image\).png](https://en.wikipedia.org/wiki/Lenna#/media/File:Lenna_(test_image).png)



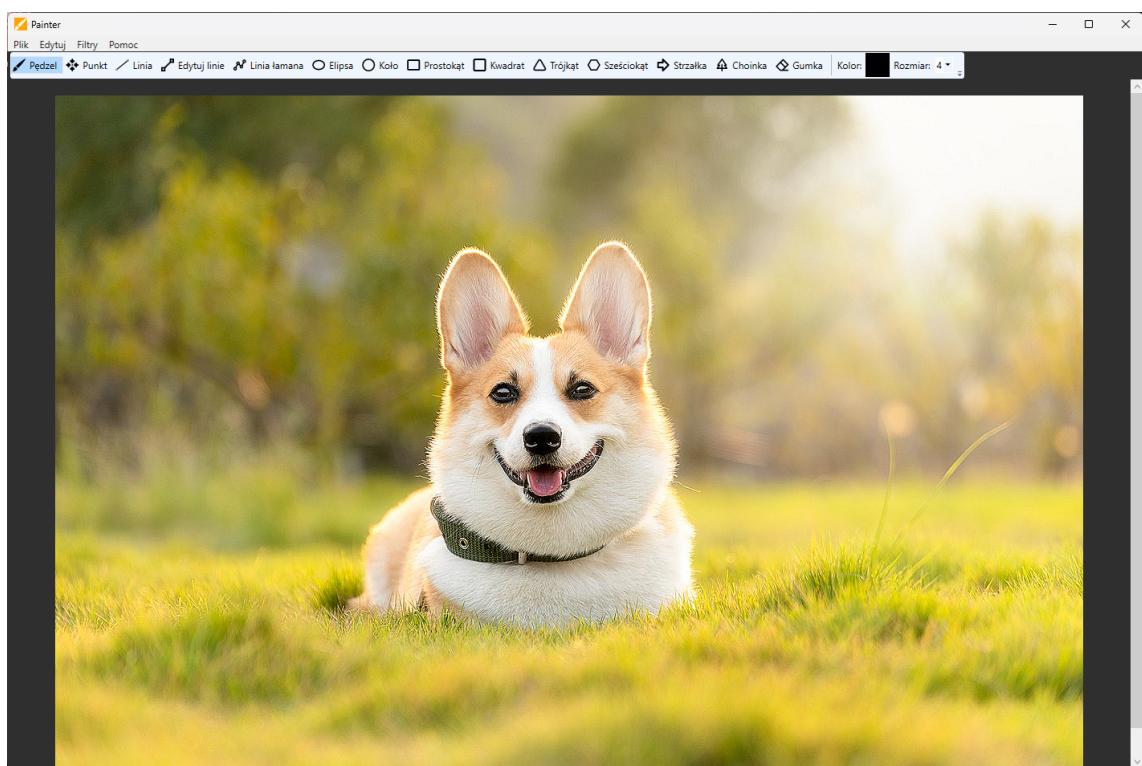
Rysunek 13: Test filtrów - skala szarości

Grafika: <https://pixabay.com/photos/dog-corgi-cute-animal-4988985/>



Rysunek 14: Test filtrów - odwrócenie kolorów

Grafika: <https://pixabay.com/photos/dog-corgi-cute-animal-4988985/>



Rysunek 15: Test filtrów - filtr macierzowy wyostrzający
Grafika: <https://pixabay.com/photos/dog-corgi-cute-animal-4988985/>

4 Podsumowanie i wnioski

4.1 Podsumowanie

W stworzonym projekcie udało się zaimplementować wszystkie wymagane funkcjonalności, a nawet rozszerzyć program o kilka dodatkowych. Zastosowanie framework'u WPF oraz zestawu klas i narzędzi w nim dostępnych w znaczącym stopniu uprościło proces tworzenia programu. Stworzony projekt ma również potencjał na dalszy rozwój.

Potencjalne scieżki rozwoju projektu:

- Dodanie dodatkowych narzędzi
- Dodanie dodatkowych filtrów
- Obsługa wartstw
- Możliwość zmiany koloru tła
- Możliwość wycinania obiektów z grafiki
- Refaktoryzacja projektu do architektury MVVM

4.2 Wnioski

Proces tworzenia programu graficznego jest złożonym procesem, ale dzięki wykorzystaniu gotowych rozwiązań można go w znacznym stopniu uproszczyć. Prace nad projektem pozwoliły zaznajomić się z metodami tworzenia programów graficznych, oraz rozbudowały i usystematyzowały wiedzę dotyczącą języka C# oraz framework'a WPF.